IN THE CLAIMS:

1. (Currently Amended) A logically partitioned data processing system, comprising:

a plurality of logical partitions;

a plurality of operating systems, each assigned to a separate one of the plurality of logical partitions;

a plurality of assignable resources, wherein each of the plurality of assignable resources is assigned to one of the plurality of logical partitions; and

a hypervisor, wherein the hypervisor emulates shared resources and provides a virtual copy of the shared resources to each of the plurality of logical partitions.

2. (Original) The logically partitioned data processing system as recited in claim 1, wherein the shared resources comprise an operator panel.

3. (Original) The logically partitioned data processing system as recited in claim 1, wherein the shared resources comprise a system console.

4. (Previously Presented) A logically partitioned data processing system, comprising:

a plurality of logical partitions;

a plurality of operating systems, each assigned to a separate one of the plurality of logical partitions;

a plurality of assignable resources, wherein each of the plurality of assignable resources is assigned to one of the plurality of logical partitions;

a hypervisor, wherein the hypervisor emulates shared resources and provides a virtual copy of the shared resources to each of the plurality of logical partitions, wherein the hypervisor receives a system message from one of the plurality of operating systems, appends an operating system identity to the message to produce a new message, and sends the new message to an external data processing system.

5. (Original) The logically partitioned data processing system as recited in claim 1, wherein instructions for executing the hypervisor are contained within firmware.

6. (Original) The logically partitioned data processing system as recited in claim 5, wherein the firmware comprises a read-only memory.

7. (Original) The logically partitioned data processing system as recited in claim 5, wherein the firmware comprises a programmable read-only memory.

8. (Original) The logically partitioned data processing system as recited in claim 5, wherein the firmware comprises an erasable programmable read-only memory.

9. (Original) The logically partitioned data processing system as recited in claim 5, wherein the firmware comprises an electrically erasable programmable read-only memory.

10. (Original) The logically partitioned data processing system as recited in claim 5, wherein the firmware comprises a non-volatile random access memory.

11-24. (Cancelled)

25. (Currently Amended) A method of allocating resources in a logically partitioned data processing system, comprising:

assigning non-overlapping subsets of resources to one of a plurality of partitions;

executing a plurality of operating systems, wherein each of the plurality of operating systems is respectively assigned to one of the plurality of partitions; and

emulating a hardware resource that is shared with the plurality of operating systems, wherein the hardware resource is not included in any of the subsets of resources.

26. (Previously Presented) The method of claim 25, wherein the step of emulating further comprises:

providing an interface to the hardware resource as an emulated port device driver.

27. (Previously Presented) The method of claim 25, wherein the step of emulating further comprises:

providing a respective virtual copy of the hardware resource to each of the plurality of partitions.

28. (Currently Amended) A method of allocating resources in a logically partitioned data processing system, comprising:

assigning non-overlapping subsets of resources to one of a plurality of partitions;

executing a plurality of operating systems, wherein each of the plurality of operating systems is respectively assigned to one of the plurality of partitions;

emulating a hardware resource that is shared with the plurality of operating systems, wherein the hardware resource is not included in any of the subsets of resources;

[The method of claim 25, further comprising:]

receiving a system message from one of the plurality of operating systems;

appending an operating system identity to the message to produce a new message, and

sending the new message to an external data processing system.

29. (Previously Presented) The method of claim 25, wherein the step of emulating further comprises:

virtualizing the hardware resource by a firmware device.

30. (Previously Presented) The method of claim 29, wherein the step of virtualizing further comprises:

performing firmware calls that emulate a port device driver.